Recall: Model

- **Definition.** [Folk]
  A model is an abstract, formal, mathematical representation or description of structure or behaviour of a (software) system.

- **Definition.** (Glinz, 2008, 425)
  A model is a concrete or mental image (Abbild) of something or a concrete or mental archetype (Vorbild) for something.

Three properties are constituent:

1. **Image attribute** (Abbildungsmerkmal), i.e. there is an entity (called original) whose image or archetype the model is.
2. **Reduction attribute** (Verkürzungsmerkmal), i.e. only those attributes of the original that are relevant in the modelling context are represented.
3. **Pragmatic attribute**, i.e. the model is built in a specific context for a specific purpose.
Our Premises for Using a Software Modelling Language

(i) We want to know how the words of the language look like: Syntax.

(ii) We want to know what a word of the language means: Semantics.

→ then we can formally analyse the model, e.g., prove that the design satisfies the requirements, simulate the model, automatically generate test cases, generate code, etc.

• UML is sometimes (neutrally, or as offence) called “semi-formal”: the UML standard OMG (2011a, b) is strong on (i), but weak(er) on (ii).

("the diagram is self-explanatory", "everybody understands the diagram" — No.)

• In the lecture: study the (!) syntax, define one (!) semantics.
• Development Process
  UML is only the language for artefacts. But we'll discuss exemplarily, where in an abstract development process which means could be used.

• How to come up with a good design
  UML is only the language to write down designs. But we'll have a couple of examples.

• Artefact Management
  Versioning, Traceability, Propagation of Changes.

• Every little bit and piece of UML
  Boring. Instead we learn how to read the standard.

• Object Oriented Programming
  Interesting: inheritance is one of the last lectures.

---

• Lecturer:
  Dr. Bernd Westphal

• Support:
  Milan Vujinovic

• Homepage:
  http://swt.informatik.uni-freiburg.de/teaching/WS2015-16/sdmauml

• Time/Location:
  Tuesday, Thursday, 10:00 – 12:00 / here (building 51, room 03-026)

• Course language:
  English (slides/writing, presentation, questions/discussions)

• Presentation:
  half slides/half on-screen

• Script/Media:
  • slides with annotations on homepage, typically soon after the lecture
    • recording on ILIAS with max. 1 week delay (links on homepage)

• Break:
  • We'll have a 10 min. break in the middle of each event from now on, unless a majority objects now.

---

• You should work in groups of approx. 3

• Please submit via ILIAS (cf. homepage); paper submissions are tolerated.

• Schedule:
  Week N, Thursday, 10–12
    Lecture A1 (exercise sheet A online)
  Week N + 1, Tuesday 10–12
    Lecture A2
    Thursday 10–12
    Lecture A3
  Week N + 2, Monday, 12:00 (exercises A early submission)
    Tuesday, 10:00 (exercises A late submission)

• Rating system:
  "most complicated rating system ever"

• Admission points
  (good-will rating, upper bound)
  "reasonable proposal given student's knowledge before tutorial"

• Exam-like points
  (evil rating, lower bound)
  "reasonable proposal given student's knowledge after tutorial"

10% bonus for early submission.

• Tutorial:
  Plenary, not recorded.

• Together develop one good solution based on selection of early submissions (anonymous) — there is no "Musterlösung" for modelling tasks.

---

• Exam Admission:
  Achieving 50% of the regular admission points in total is sufficient for admission to exam. Typically, 20 regular admission points per exercise sheet.

• Exam Form:
  • oral for BSc and on special demand (Erasmus),
  • written for everybody else (if sufficiently many candidates remain).

Scores from the exercises do not contribute to the final grade.

• Exam Date:
  Remind me in early December that we need to agree on an exam date.

---

• Approach:
  The lectures is supposed to work as a lecture: spoken word + slides + discussion. It is not our goal to make any of the three work in isolation.

• Interaction:
  Absence often moaned but it takes two: please ask/comment immediately.

• Exercise submissions:
  Each task is a tiny little scientific work:
  (i) Briefly rephrase the task in your own words.
  (ii) State your claimed solution.
  (iii) Convince your reader that your proposal is a solution (proofs are very convincing).
Approach: The lectures is supposed to work as a lecture: spoken word + slides + discussion. It is not our goal to make any of the three work in isolation.

Interaction: Absence often moaned but it takes two: please ask/comment immediately.

Exercise submissions: Each task is a tiny little scientific work:

(i) Briefly rephrase the task in your own words.
(ii) State your claimed solution.
(iii) Convince your reader that your proposal is a solution (proofs are very convincing).

Example: Task: Given a square with side length \( a = 19.1 \).

What is the length of the longest straight line fully inside the square?

Submission A: Submission B:

The length of the longest straight line fully inside the square with side length \( a = 19.1 \) is 27.01 (rounded).

The longest straight line inside the square is the diagonal. By Pythagoras, its length is \( \sqrt{a^2 + a^2} \). Inserting \( a = 19.1 \) yields 27.01 (rounded).


